Operator Splitting Methods Combined with Multi-grid Methods

Jűrgen Geiser

EMA University of Greifswald, Institute of Physics

Felix-Hausdorff-Str. 6, D-17489 Greifswald, Germany

geiser@mathematik.hu-berlin.de

***Abstract*-In this article a new approach is considered for implementing multi-grid methods in iterative operator splitting methods for differential equations. The underlying idea is to embed fast multi-grid methods to accelerate the iterative splitting schemes. We concentrate on convection–diffusion–reaction equations and treat a multi-scale problem. The equations are spatially discretised with finite difference or finite volume methods. The main problem of iterative solvers is their neglect of multi-scale physics: different scales are not resolved on different spatial scales. Here we apply multi-grid methods to obtain the resolution of the optimal scale, which can be solved by an iterative splitting scheme. We discuss the embedding of such spatial- and time-scale methods and their underlying analysis. Numerical examples are given to verify the numerical schemes.**

Keywords*- *Numerical Analysis; Operator Splitting Method; Initial Value Problems; Iterative Solver Method; Stability Analysis; Convection–Diffusion–Reaction Equations; Multi-Grid Methods

AMS subject classifications. 35K15 35K57 47F05 65M60 65N30

I. INTRODUCTION

Our motivation is to solve multi-scale problems that occur, for example, in transport–reaction problems in porous media.

In the last years, the interest in numerical simulations of multi-scale problems that can be used to model different physical behaviors, e.g., potential damage events, has significantly increased, but the adaptation of this methodology to such problems is delicate, see [39, 40].

We will concentrate on simplified models and give an account of the ideas needed to adapt the underlying splitting schemes, see [15]. While these splitting schemes worked well for single scale problems, we have to embed multi-grid ideas or multi-step ideas into these schemes to extend them to be multi-scale solvers. We will deal with iterative splitting methods and how to decompose such scale problems and accelerate the convergence rates with multi-scale methods, so that we can overcome such scaling problems.

Moreover the embedding part is important: while dealing with multi-grid methods, we have to extend the underlying splitting analysis. We will discuss the needed algorithmic ideas.

The novelty is to cheaply embed multi-grid and multi-step methods and apply only cheap iterative schemes to achieve higher order results.

In general, we discuss an elegant way of embedding and survey recent methods for multi-scale problems with respect to iterative splitting schemes.

In the following, we describe our model problem. The model equation for the multi-scale equations are given as coupled partial differential equations.

We concentrate on a far-field model for a plasma reactor, see [16] and [30], and assume a continuum flow and that the transport equations can be treated with a convection–diffusion–reaction equation, due to a constant velocity field:

$$\frac{\partial c}{\partial t} + \nabla \cdot Fc = 0, \text{in} \Omega \times [0, t] \tag{1}$$

$$F = v - D\nabla,$$

$$c(x, t) = c_0(x), \text{on } \Omega, \tag{2}$$

$$\frac{\partial c(X,t)}{\partial n} = 0, \text{on } \partial\Omega \times [0, t] \tag{3}$$

where c is the particle density of the ionised species, F is the flux of the species, v is the flux velocity through the chamber and porous substrate, which is influenced by the electric field, and D is the diffusion matrix. The initial value is given as c0 and we assume a Neumann boundary condition.

The aim of this paper is to present a novel iterative splitting method that embeds multi-scale methods for spatially discretised transport–reaction equations.

This paper is organized as follows: In Section Ⅱ, we present the underlying splitting methods. The stability analysis of the embedded multi-grid schemes with the iterative schemes is given in Section Ⅲ. The assembling and the algorithms for the embedded multi-grid method are given in Section Ⅳ. Numerical verifications are given in Section Ⅴ. In Section Ⅵ, we briefly summarize our results.

II. ITERATIVE SPLITTING METHODS

Iterative splitting methods were developed in the early 90s, see [28, 41].

The idea is to decouple the equations into two or more equations to save computational time, without changing the previous results. We obtain inhomogeneous partial differential equations and solve them with the appropriate methods.

We consider the following differential equation problem, where the operators A and B are spatially discretised:

$$\frac{\partial u}{\partial t} = Au + Bu \tag{4}$$

where the initial conditions are un = u(tn). The solutions correspond via the spatial discretisation by u(t)

= (u1 (t), . . . , um (t))T = (c(x1 , t), . . . , c(xm , t)T and m is the number of spatial discretisation points. The operators

A and B are matrices corresponding to the discretised convection and diffusion operators in (1) and their boundary conditions. Hence, they can be considered as bounded operators with a sufficient large spatial step $\Delta x > 0$.

In the following we discuss the different methods based on the following definition:

Definition 1. We define a sequential method as a serial method, while each step can only be done successively (e.g., each processor has to wait till the results of the previous computation are done). We define a parallel method as a multitasking method, which can be executed one piece at a time on many different processing devices, and then put back together again at the end to get the correct result. (e.g., each processor can independently compute a result).

A. The Sequential Iterative Splitting Method

The classical sequential operator splitting methods, known as Lie-splitting or Strang–Marchuk splitting, have several drawbacks besides their benefits, see [38, 31]. For instance, for non-commuting operators there might be a very large constant in the splitting error which requires the use of an unrealistically small time step. Also, splitting the original problem into different sub-problems with one operator, i.e., neglecting the other components, is physically questionable.

In order to avoid these problems, one can use the iterative operator splitting method on the interval [0, T]. This algorithm is based on an iteration with a fixed splitting discretisation step-size τ . On every time interval $[t^n, t^{n+1}]$ the method solves the following sub-problems consecutively for $i = 0, 2, \dots 2m$.

$$\frac{\partial u_i(t)}{\partial t} = Au_i(t) + Bu_{i-1}(t), \text{with } u_i(t^n) = u^n, t \in (t^n, t^{n+1}), \quad (5)$$

$$\frac{\partial u_{i+1}(t)}{\partial t} = Au_i(t) + Bu_{i+1}(t), \text{with } u_{i+1}(t^n) = u^n, \ t \in (t^n, t^{n+1}), \quad (6)$$

where u^n is the known split approximation at the time level $t = t^n$ (see [9]).. This algorithm is an iterative method which involves in each step both operators A and B. Hence, there is no real separation of the different physical processes in these equations.

Remark 1. For the presented iterative splitting method, we have a serial algorithm and we cannot use parallel methods in this version. For the sake of efficiency, we will modify this method to enable parallelisation.

B. The Parallel Iterative Splitting Method

To parallelise the standard iterative splitting method, see Section Ⅱ-A, we propose a decoupled version.

The iterative scheme can be done in a Jacobian form, so that the two parts of the algorithm can be computed independently.

In the next iteration steps, we use the previous solutions and improve them.

We apply the parallel iterative splitting method for i = 0, 2, . . . 2m, and we have:

$$\frac{\partial u_i(t)}{\partial t} = Au_i(t) + Bu_{i-1}(t), \text{with } u_i(t^n) = u^n, \ t \in (t^n, t^{n+1}) \quad (7)$$

$$u_0(t^n) = u^n, u_{i-1}(t) = 0,$$

$$\frac{\partial u_{i+1}(t)}{\partial t} = Au_{i-2}(t) + Bu_{i+1}(t), \text{with } u_{i+1}(t^n) = u^n, t \in (t^n, t^{n+1}), \quad (8)$$

$$u_{-2}(t) = 0$$

where u^n is the known split approximation at the time level *t=tn*. This algorithm is an iterative method where each step involves both Operators A and B. Hence, there is no real separation of the different physical processes.
The first iterative steps are:

$$\frac{\partial u_0(t)}{\partial t} = Au_0(t), \text{with } u_0(t^n) = u_0, \quad (9)$$

$$\frac{\partial u_1(t)}{\partial t} = Bu_1(t), \quad (10)$$

$$\text{with } u_1(t^n) = u^n, \quad (11)$$

and

$$\frac{\partial u_2(t)}{\partial t} = Au_2(t) + Bu_1(t), \text{with } u_2(t^n) = u_0,$$

$$\frac{\partial u_3(t)}{\partial t} = Au_0(t) + Bu_3(t),$$

$$\text{with } u_3(t^n) = u^n, \quad (12)$$

Remark 2. The effect of the parallelisation is to obtain the accuracy of the sequential iterative splitting method with one more iterative step.

C. The Multi-grid Algorithm

To describe the multi-grid method we first implement the two-grid method. Afterwards, we extend it recursively to the multi-grid method. The smoother grid Level l is denoted by Sl. The two-grid method is defined as follows:

$$M_l^{ZG} := S_l^{v_2} \, M_l^{ZGG} \, S_l^{v_1} \quad (13)$$

where *v1* denotes the pre-smoothing steps and *v2* denotes the post-smoothing steps. The correction M_l^{ZGG} on the coarse grid is defined by:

$$M_l^{ZGG} := 1 - pA_{l-1}^{-1} rA_l \quad (14)$$

The passage to the multi-grid method is done in such a way that Al−1 of the coarse grid in Equation (14) is not inverted exactly, but the two-grid method is invoked γ-times to solve the equation systems at the grid level l − 1. The system of equations is only solved on the coarsest grid.

The multi-grid method is defined as:

$$M_0^{MG} := 0, \quad (15)$$

$$M_l^{MG} := M_l^{ZG} \quad (16)$$

$$M_l^{MG} := M_l^{ZG} + S_l^{v_2} p\left(M_{l-1}^{MG}\right)^{\gamma} A_{l-1}^{-1} rA_l S_l^{v_1} \quad (17)$$

where ν1 denotes the pre-smoothing steps and ν2 the post-smoothing steps. The corrections M_l^{ZGG} of the coarse grid are:

$$M_l^{MGG} := 1 - p(I - (M_{l-1}^{MG})^{\gamma}) A_{l-1}^{-1} \gamma A_l. \quad (18)$$

This approach is called the multi-grid method. For a choice of γ =1 one speaks of a V-cycle, for γ = 2 it is a W-cycle.

The multi-grid algorithm is given by

Algorithm 1 Linear multi-grid cycle

$\boldsymbol{MG}\ (x_l, b_l, l)$

{

$\quad$ $\boldsymbol{if}(\boldsymbol{l} == \boldsymbol{0})$

$\quad$ {

$\quad$ $x_0 = A_0^{-1} b_0;$ $\quad$ ***exact solving on coarse grid.***

$\quad$ }

$\quad$ ***else***

$\quad$ {

$\quad\quad$ $x_1 = S^{v_1}(x_l, b_l);$ $\quad$ v_1 ***pre − smoothing steps.***

$\quad\quad$ $b_{l-1} = r b_l;$ ***defect restricted on next coarser grid.***

$\quad\quad$ $x_{l_1} = 0;$

$\quad\quad$ **for** (*i=0; i < γ; i++*)

$\quad\quad$ {

$\quad\quad\quad$ $c_{l-1} = 0;$

$\quad\quad\quad$ $MG(c_{l-1}, b_{l-1}, l-1);$ $\quad$ ***γ − fold recursive invoking of correction of coarse grid***

$\quad\quad\quad$ $x_{l-1} = x_{l-1} + c_{l-1};$

$\quad\quad\quad$ $b_{l-1} = b_{l-1} - A_{l-1} c_{l-1};$

$\quad\quad$ }

$\quad\quad$ $c_l = p x_{l-1};$ **interpolaton of correction of the next coarse grid.**

$\quad\quad$ $x_l = x_l + c_l;$

$\quad\quad$ $b_l = b_l - A_l c_l;$

$\quad\quad$ $x_l = S^{v_2}(x_l, b_l);$ v_2 **post − smoothing steps.**

$\quad$ }

}

The refinement is done using the strategy of [18]. By assuming a linear effort for the smoothers as well as for the grid-transfer operators, one obtains a linear effort for Algorithms 1, if $\gamma \leq 3$, cf. [18]. The proofs of convergence for the W-cycle were given in [17]. The topic of convergence will not be further discussed, for an overview we refer to [44].

Subsequently, the multi-grid cycles are illustrated in Fig. 1. For a further treatment of the topic of multi-grid methods, we refer to [2, 17, 18].

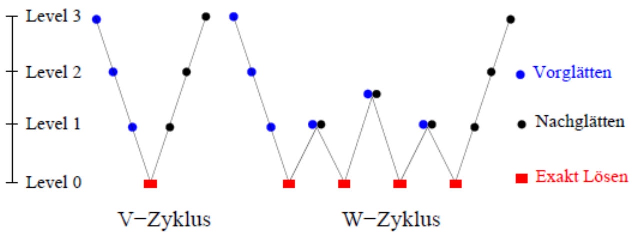

Fig. 1 Multi-grid cycles: V- and W-cycle

D. Iterative Splitting Method with Embedded Multi-grid Method

The following algorithm is based on embedding the multi-grid method into the operator splitting method. The iteration with fixed splitting discretisation step- size τ . On the time interval [*tn* , *tn+1*] we solve the following sub-problems consecutively for *i = 0, 2, . . . 2m.* (cf. [24] and [9].)

$$\frac{\partial c_i(t)}{\partial t} = A c_i(t) + P^{l_A - l_B} + B c_{i-1}(t), \text{with } c_i(t^n) = c^n, \tag{19}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A - l_B} A c_i(t) + B c_{i+1}(t), \text{with } c_{i+1}(t^n) = c^n, \tag{20}$$

where $c_0\ (t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at $t = t^n$. We assume A is the fine spatial discretised operator on level *lA* , where B is the coarse discretised operator on level *lB* .

The operators are coupled by the restriction and prolongation operators:

$$A_{\text{coarse}} = P^{l_A - l_B} A, \tag{21}$$

$$B_{\text{fine}} = P^{l_A - l_B} B, \tag{22}$$

where R is the restriction and P the prolongation operator.

Algorithm 2 We assume two spatial operators A, B, which are discretised by finite difference of finite element methods.

We solve the time-discretisation of our equations

$$\frac{\partial c_i(t)}{\partial t} = A c_i(t) + P^{l_A - l_B} B c_{i-1}(t), \text{with } c_i(t^n) = c^n, \tag{23}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A - l_B} A c_i(t) + B c_{i+1}(t), \text{with } c_{i+1}(t^n) = c^n, \tag{24}$$

where $c0\ (t^n) = c^n$, $c{-1} = 0$ and c^n and *cn* is the known split approximation at time-level $t = t^n$ by integration with respect to time and obtain

$$(I - A) c_i(t^{n+1}) = c_i(t^n) + P^{l_A - l_B} B c_{i-1}(t^{n+1}), \quad \text{with } c_i(t^n) = c^n \tag{25}$$

$$(I - B) c_{i+1}(t) = c_{i+1}(t^n) + R^{l_A - l_B} A c_i(t^{n+1}), \quad \text{with } c_{i+1}(t^n) = c^n, \tag{26}$$

We have the multi-grid equations:

$$L_i c_i(t^{n+1}) = c_i(t)^n + P^{l_A - l_B} B c_{i-1}(t^{n+1}), \quad \text{with } c_i(t^n) = c^n, \tag{27}$$

$$L_{i+1} c_{i+1}(t) = c_{i+1}(t^n) + R^{l_A - l_B} A c_i(t^{n+1}), \quad \text{with } c_{i+1}(t^n) = c^n, \tag{28}$$

E. The Waveform Relaxation Method

A further method to solve large coupled differential equations is the waveform relaxation scheme.

The iterative method was discussed in [41] and can be done either with Gaussian or Jacobian form.

We deal with the following ordinary differential equation or assume a semi-discretised partial differential equation:

ut = f (u, t), in (0, T) , u(0) = v0

where *u = (u1 , . . . , um)t* and *f (u, t) = (f1 (u, t), . . . , fm (u, t))*^{t}.

We apply the waveform relaxation method for i = 0, 1, . . . m and have:

$$\frac{\partial u_{1,i}(x,t)}{\partial t} = f_1(u_{1,i}, u_{2,i-1}, \ldots, u_{m,i-1}) \quad \text{with } u_{1,i}(t^n) = u_1(t^n) \tag{29}$$

$$\frac{\partial u_{2,i}(x,t)}{\partial t} = f_2(u_{1,i-1}, u_{2,i}, u_{3,i-1}, \ldots, u_{m,i-1}) \quad \text{with } u_{2,i}(t^n) = u_2(t^n) \tag{30}$$

…

$$\frac{\partial u_{m,i}(x,t)}{\partial t} = f_2\left(u_{1,i-1}, \dots, u_{m-1,i-1}, u_{m,i}\right)$$

$$\text{with } u_{m,i}(t^n) = u_{m,1}(t^n) \quad (31)$$

where for the initialisation of the first step we put $u_{1,-1}(t) = u_1(t^n), \dots, u_{m,-1}(t) = u_m(t^n)$.

We reduce to two equations and reformulate the method to our iterative splitting methods.

So we consider

$$\frac{\partial u_1}{\partial t} = f_{11}(u_1, t) + f_{12}(u_2, t), \text{in } (0, T) \quad (32)$$

$$\frac{\partial u_2}{\partial t} = f_{21}(u_1, t) + f_{22}(u_2, t), \text{in } (0, T) \quad (33)$$

$$u(0) = v_0 \quad (34)$$

where $u = (u_1, u_2)^t$.

Our notation for the operator equation is

$$\frac{\partial u}{\partial t} = A(u) + B(u), \text{in } (0, T) \quad (35)$$

$$u(0) = v_0 \quad (36)$$

Where

$$A(u) = \begin{pmatrix} f_{11}(u_1) \\ f_{21}(u_1) \end{pmatrix} \quad (37)$$

$$B(u) = \begin{pmatrix} f_{12}(u_2) \\ f_{22}(u_2) \end{pmatrix} \quad (38)$$

The iterative splitting method as waveform relaxation method written for

$i = 0, 1, \dots m$ as:

$$\frac{\partial u_i}{\partial t} = A\left(u_{1,i}, u_{2,i-1}\right) + B\left(u_{1,i-1}, u_{2,i}\right) \quad (39)$$

$$\text{with } u_{1,i}(t^n) = u_1(t^n) \text{ and } u_{2,i}(t^n) = u_2(t^n)$$

where for the initialisation of the first step we put $u_{1,-1}(t) = u_1(t^n), u_{2,-1}(t) = u_2(t^n)$.

Convergence analysis In the following we formulate the iterative splitting method as a waveform relaxation method and apply the proof techniques of those relaxation schemes.

The waveform relaxation method is given by

$$\frac{du_i}{dt} = Pu_i + Qu_{i-1} + f, \quad (40)$$

$$u_i(t^n) = u(t^n), \quad (41)$$

where A = P + Q, e.g., P is the diagonal part of A (Jacobi method).

Here the splitting method is done abstractly with respect to the Matrix A. The method is considered an effective solver method with respect to the underlying matrices.

The reformulation of the iterative operator splitting method is given by

$$\frac{du_i}{dt} = Pu_i + Qu_{i-1} + f, \quad (42)$$

$$u_i(t^n) = u(t^n), \quad (43)$$

$$\frac{du_{i+1}}{dt} = Pu_i + Qu_{i+1} + f, \quad (44)$$

$$u_{i+1}(t^n) = u(t^n), \quad (45)$$

where P, Q are matrices given by spatial discretisation, e.g., P is the convection part of Q the diffusion part.

But we can also make an abstract decomposition, taking into account that A = P + Q, where P is the matrix with small eigenvalues and Q is the matrix with large eigenvalues.

Theorem 1 We have the iterative operator splitting methods given as the following outer and inner iterations.

Outer Iteration:

$$\frac{dU_i}{dt} = PU_i + QU_{i-1} + F, \quad (46)$$

$$U_i(t^n) = U(t^n), \quad (47)$$

where P and Q are the diagonal and outer-diagonal matrices of the splitting methods.

Inner Iteration:

$$\frac{du_j}{dt} = Pu_j + Qu_{j-1} + f, \quad (48)$$

$$u_j(t^n) = u(t^n), \; j = 1, \dots, J \quad (49)$$

$$\frac{du_k}{dt} = Pu_J + Qu_k + f, \quad (50)$$

$$u_k(t^n) = u(t^n), \; k = 1, \dots, K \quad (51)$$

We have a convergent scheme for K bounded in a Banach-space:

$$\rho(K) := \lim_{\kappa \to \infty} \|K^\kappa\|^{1/\kappa} \quad (52)$$

where $\rho(K) \leq 1$ is the spectral radius of K and is given by the variation of constants of *P and Q*.

Proof. The outer iteration is given by

$$\frac{dU_i}{dt} = PU_i + QU_{i-1} + F, \quad (53)$$

$$U_i(t^n) = U(t^n), \quad (54)$$

where P and Q are the diagonal and outer-diagonal matrices of the splitting methods, which can be solved by the variation of constants.

We introduce the linear integral operator

$$KU(t) = \int_0^t \exp\left((t-s)P\right) QU(s)ds, \quad (55)$$

And we have

$$U_i = KU_{i-1} + \emptyset, \quad (56)$$

$$\emptyset = \exp(tP)U(t^n) + \int_0^t \exp\left((t-s)P\right) F(s)ds, \quad (57)$$

For the convergence it is sufficient to show that K is bounded in a Banach-space:

$$\rho(K) := \lim_{\kappa \to \infty} \|K^\kappa\|^{1/\kappa} \quad (58)$$

where $\rho(K) \leq 1$ is the spectral radius of K

This is given by the bounded operators, see [33, 34].

III. ERROR ANALYSIS: STABILITY THEORY FOR THE ITERATIVE SPLITTING METHOD WITH MULTI-GRID METHODS

The following algorithm is based on embedding the multi-grid method in the operator splitting method. The iteration with fixed splitting discretisation is step-size τ. On the time interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for i = 0, 2, . . . 2m. (cf. [24, 9].)

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + P^{l_A-l_B}Bc_{i-1}(t), \text{with } c_i(t^n) = c^n, \tag{59}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A-l_B}Ac_i(t) + Bc_{i+1}(t), \text{with } c_{i+1}(t^n) = c^n, \tag{60}$$

where $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at time-level $t = t^n$. We assume *A* to be the fine spatial discretised operator on level *lA* and *B* to be the coarse discretised operator on level *lB*.

The operators are coupled by the restriction and prolongation operators:

$$A_{coarse} = R^{l_A-l_B}A, \tag{61}$$

$$B_{fine} = P^{l_A-l_B}B, \tag{62}$$

where R is the restriction and P the prolongation operator.

Theorem 2 Let us consider the abstract Cauchy problem in a Banach space X

$$\partial_t c(t) = Ac(t) + P^{l_A-l_B}Bc(t), \quad 0 < t \le T$$
$$c(0) = c_0 \tag{63}$$

where A, $P^{lA-lB}B$, $A + P^{lA-lB}B : X \to X$ are given linear operators being generators of the C_0-semigroup and $c_0 \in X$ is a given element. Then the iteration process (59)–(60) is convergent and the rate of convergence is of higher order.

Proof. We assume $A + P^{lA-lB}B \in L(X)$ and assume a generator of a uniformly continuous semi-group, hence the problem (63) has a unique solution $c(t)=exp((A + P^{lA-lB}B)t)c_0$.

Let us consider the Iteration (59)–(60) on the sub-interval *[tn, tn+1]*. For the local error function *ei(t) = c(t) − ci (t)* we have the relations

$$\partial_t e_i(t) = Ae_i(t) + P^{l_A-l_B}Be_{i-1}(t), \quad t \in (t^n, t^{n+1}]$$
$$e_i(t^n) = 0, \tag{64}$$

And

$$\partial_t e_{i+1}(t) = R^{l_A-l_B}Ae_i(t) + Be_{i+1}(t), \quad t \in (t^n, t^{n+1}]$$
$$e_{i+1}(t^n) = 0, \tag{65}$$

for *m* =0, 2, 4, . . ., with *e0(0) = 0* and *e−1(t) = c(t)*. In the following we use the notations X2 for the product space X × X endowed with the norm

$\|(u,v)\| = \max\{\|u\|, \|v\|\}\ (u, v \in X)$ The elements

$\mathcal{E}_i(t), F_i(t) \in X^2$ and the linear operator

$A : X^2 \longrightarrow X^2$ are defined as follows:

$$\varepsilon_i(t) = \begin{bmatrix} e_i(t) \\ e_{i+1}(t) \end{bmatrix}, F_i(t) = \begin{bmatrix} P^{l_A-l_B}Be_{i-1}(t) \\ 0 \end{bmatrix},$$

$$A = \begin{bmatrix} A & 0 \\ R^{l_A-l_B}A & B \end{bmatrix} \tag{66}$$

Then using the Notation (66), the Relations (64) and (65) can be written in the form

$$\partial_t \varepsilon_i(t) = A\varepsilon_i(t) + F_i(t), t \in (t^n, t^{n+1}], \varepsilon_i(t^n) = 0. \tag{67}$$

Due to our assumptions, A is a generator of the one-parameter C_0-semi-group (exp *At)* $t \ge 0$, hence using the variations of constants formula, the solution to the abstract Cauchy problem (67) with homogeneous initial condition can be written as

$$\varepsilon_i(t) = \int_{t^n}^{t} \exp(A(t-s))\, F_i(s)ds, t \in [t^n, t^{n+1}] \tag{68}$$

(See, e.g., [8]) Hence, using the notation

$$\|\varepsilon_i\|_\infty = \sup_{t\in[t^n,t^{n+1}]}\|\varepsilon_i(t)\| \tag{69}$$

We have

$$\|\varepsilon_i\|(t) \le \|F_i\|_\infty \int_{t^n}^{t}\|\exp(A(t-s))\|ds =$$
$$\|B\|\|e_{i-1}\| \int_{t^n}^{t}\|\exp(A(t-s))\|ds, t \in [t^n, t^{n+1}]. \tag{70}$$

Since (A(t))t ≥ 0 is a semi-group, the so called growth estimation

$$\|\exp(At)\| \le K\exp(wt), t \ge 0, \tag{71}$$

holds with some numbers K ≥ 0 and ω ∈ IR, cf. [8].

– Assume that (A(t))t ≥ 0 is a bounded or an exponentially stable semi-group, i.e., (71) holds with some ω ≤ 0. Then obviously the estimate

$$\|\exp(At)\| \le K, t \ge 0, \tag{72}$$

holds, and hence, on the basis of (70), we have the relation

$$\|\varepsilon_i\|(t) \le K\|P^{l_A-l_B}B\|T_n\|e_{i-1}\|, t \in [t^n, t^{n+1}] \tag{73}$$

– Assume that (exp At)t ≥ 0 has exponential growth with some ω > 0. Using (71), we have

$$\int_{t^n}^{t}\|\exp(A(t-s))\|\, ds \le K_w(t), t \in [t^n, t^{n+1}] \tag{74}$$

Where

$$K_w(t) = \frac{K}{w}(\exp(w(t-t^n)) - 1), t \in [t^n, t^{n+1}] \tag{75}$$

Hence

$$K_w(t) \le \frac{K}{w}(\exp(wT_n) - 1) = K_{T_n} + O(T_n^2). \tag{76}$$

The Estimations (73) and (76) result in

$$\|\varepsilon_i\|_\infty = K\|P^{l_A-l_B}\|\|B\|T_n\|e_{i-1}\| + O(T_n^2) \tag{77}$$

Taking into account the definition of Ei and of the norm k · k∞, we obtain

$$\|e_i\| = K\|P^{l_A-l_B}\|\|B\|T_n\|e_{i-1}\| + O(T_n^2) \tag{78}$$

And hence

$$\|e_{i+1}\| = K_1 T_n^2 \|e_{i-1}\| + O(T_n^3), \qquad (79)$$

which proves our statement.

Operator-splitting method with embedded Jacobian Newton iterative method: Newton's method is used to solve the nonlinear parts of the iterative operator-splitting method, see the linearisation techniques in [24, 25]. We apply the iterative operator-splitting method and obtain

$$F_1(u_i) = \partial_t u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0,$$
$$\text{with } u_i(t^n) = c^n,$$
$$F_2(u_{i+1}) = \partial_t u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0,$$
$$\text{with } u_{i+1}(t^n) = c^n,$$

where the time step is$\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \ldots, 2m + 1$. $c_0(t) = 0$ is the starting solution and cn is the known split approximation at time level $t = t^n$. The results of the methods are $c(t^{n+1}) = u2m+2(t^{n+1}$. The splitting method with embedded Newton's method is given by

$$\begin{aligned} u_i^{(k+1)} = u_i^{(k)} - D\left(F_1\left(u_i^{(k)}\right)\right)^{-1} & \left(\partial_t u_i^{(k)} - A\left(u_i^{(k)}\right)u_i^{(k)}\right. \\ & \left. - B\left(u_{i-1}^{(k)}\right)u_{i-1}^{(k)}\right), \text{with } D\left(F_1\left(u_i^{(k)}\right)\right) \\ & = -\left(A\left(u_i^{(k)}\right) + \frac{\partial A\left(u_i^{(k)}\right)}{\partial u_i^{(k)}} u_i^{(k)}\right), \end{aligned}$$

$$\text{and } k = 0,1,2,\ldots,K, \text{with } u_i(t^n) = c^n,$$

$$\begin{aligned} u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D\left(F_2\left(u_{i+1}^{(l)}\right)\right)^{-1} & \left(\partial_t u_{i+1}^{(l)} - A\left(u_i^{(k)}\right)u_i^{(k)}\right. \\ & \left. - B\left(u_{i+1}^{(k)}\right)u_{i+1}^{(k)} c_2^{(l)}\right), \end{aligned}$$

$$\text{with } D\left(F_2\left(u_{i+1}^{(l)}\right)\right) = -\left(B\left(u_{i+1}^{(l)}\right) + \frac{\partial B\left(u_{i+1}^{(l)}\right)}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}\right),$$

$$\text{and } l = 0,1,2,\ldots,L, \text{with } u_i + 1(t^n) = c^n.$$

Remark 3. For the iterative operator-splitting method with Newton's method we have two iteration procedures. The first iteration is Newton's method for computing the solution of the nonlinear equations; the second iteration is the iterative splitting method, which computes the resulting solution of the coupled equation systems. The embedded method is used for strong nonlinearities.

IV. ASSEMBLING OF THE SPLITTING METHODS WITH THE EMBEDDED MULTI-GRID METHOD

In the following we discuss the assembling of our methods.

A. Crank-Nicolson and Two-grid Method

In the following we apply the iterative splitting method, with a Crank–Nicolson method in time, a second order finite difference method in space, and an underlying two-grid method as solver.

We apply the Crank-Nicolson scheme separately to the two equations obtained by the two steps of the iterative operator splitting method after the discretisation of the 2D heat equation. For the ith step of the iterative operator splitting method, as interpolation operator we choose the bilinear interpolation given by

$$v_{2i,2j}^h = v_{ij}^{2h}$$
$$v_{2i+1,2j}^h = \frac{1}{2}\left(v_{2j}^{2h} + v_{i+1,j}^{2h}\right),$$
$$v_{2i,2j+1}^h = \frac{1}{2}\left(v_{ij}^{2h} + v_{i,j+1}^{2h}\right)$$
$$v_{2i+1,2j+1}^h = \frac{1}{4}\left(v_{ij}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}\right),$$
$$0 \le i, j, \le \frac{n}{2} - 1.$$

where the superscript *h* is associated with the fine grid and 2*h* with the coarse grid.

Step *i:*

$$\begin{aligned} \frac{u_{i.j.k}^{n+1} - u_{i,j,k}^n}{\Delta t} = \frac{1}{2}\Bigg[\Bigg(D_1 & \frac{u_{i-1,j,k}^{n+1} - 2u_{i,j,k}^{n+1} + u_{i+1,j,k}^{n+1}}{h_x^2} \\ & + PD_2 \frac{u_{i-1,j-1,k-1}^{n+1} - 2u_{i-1,j,k-1}^{n+1} + u_{i-1,j+1,k-1}^{n+1}}{h_y^2}\Bigg) \\ & + \Bigg(D_1 \frac{u_{i-1,j,k}^{n} - 2u_{i,j,k}^{n} + u_{i+1,j,k}^{n}}{h_x^2} \\ & + PD_2 \frac{u_{i-1,j-1,k-1}^{n} - 2u_{i-1,j,k-1}^{n} + u_{i-1,j+1,k-1}^{n}}{h_y^2}\Bigg)\Bigg] \end{aligned}$$

$$= \frac{D_1}{2}\delta_x^2 u_{i,j,k}^{n+1} + \frac{D_2}{2}P\delta_y^2 u_{i-1,j,k-1}^{n+1} + \frac{D_1}{2}\delta_x^2 u_{i,j,k}^{n} + \frac{D_2}{2}P\delta_y^2 u_{i-1,j,k-1}^{n}$$

$$\begin{aligned} \Leftrightarrow Pu_{i-1,j-1,k-1}^{n+1} & \left(\frac{-D_2\Delta t}{2h_y^2}\right) + u_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1} \\ & + u_{i,j,k}^{n+1}\frac{D_1\Delta t}{h_x^2} + Pu_{i-1,j,k-1}^{n+1}\frac{D_2\Delta t}{h_y^2} \\ & + Pu_{i-1,j+1,k-1}^{n+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right) + u_{i+1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) \\ & = \frac{1}{2}\left(D_1\delta_x^2 u_{i,j,k}^n + D_2 P\delta_y^2 u_{i-1,j,k-1}^n\right) + u_{i,j,k}^n \end{aligned}$$

$$\begin{aligned} \Leftrightarrow\ & u_{i-1,j-1,k}^{n+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right) + u_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1} \\ & + u_{i,j,k}^{n+1}\frac{D_1\Delta t}{h_x^2} + \frac{1}{2}\left(u_{i-1,j-1,k}^{n+1} + u_{i-1,j+1,k}^{n+1}\right)\frac{D_2\Delta t}{h_y^2} \\ & + \frac{1}{4}\left(u_{i-1,j+1,k}^{n+1} + u_{i,j+1,k}^{n+1} + u_{i-1,j+2,k}^{n+1} + u_{i,j+2,k}^{n+1}\right)\left(\frac{-D_2\Delta t}{2h_y^2}\right) \\ & + u_{i+1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) \\ & = \frac{1}{2}D_1\delta_x^2 u_{i,j,k}^n + \frac{1}{2}D_2 P\frac{u_{i-1,j-1,k-1}^n - 2u_{i-1,j,k-1}^n + u_{i-1,j+1,k-1}^n}{h_y^2} \\ & + u_{i,j,k}^n \end{aligned}$$

$$\begin{aligned} \Leftrightarrow\ & u_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1}\left(1 + \frac{D_1\Delta t}{h_x^2}\right) + u_{i-1,j,k}^{n+1}\frac{D_2\Delta t}{2h_y^2} \\ & + \frac{1}{4}\left(u_{i-1,j+1,k}^{n+1} + u_{i,j+1,k}^{n+1} + u_{i-1,j+2,k}^{n+1} + u_{i,j+2,k}^{n+1}\right)\left(\frac{-D_2\Delta t}{2h_y^2}\right) \\ & + u_{i+1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) \\ & = \frac{1}{2}D_1\delta_x^2 u_{i,j,k}^n + \frac{1}{2}\frac{D_2}{h_y^2}\Bigg[u_{i-1,j-1,k}^n - 2\,\frac{1}{2}\left(u_{i-1,j-1,k}^n + u_{i-1,j+1,k}^n\right) \\ & \qquad + \frac{1}{4}\begin{pmatrix} u_{i-1,j+1,k}^n \\ +u_{i,j+1,k}^n + u_{i-1,j+2,k}^n + u_{i,j+2,k}^n \end{pmatrix}\Bigg] \end{aligned}$$

$$\Leftrightarrow u_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1}\left(1 + \frac{D_1\Delta t}{h_x^2}\right) + u_{i-1,j+1,k}^{n+1}\left(\frac{D_2\Delta t}{2h_y^2}\right)$$

$$+u^{n+1}_{i,j+2,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i-1,j+1,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i,j+1,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)$$

$$+u^{n+1}_{i-1,j+2,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}D_1\delta_x^2u^n_{i,j,k}+\frac{1}{2}\frac{D_2}{h_y^2}\left[-\frac{1}{2}u^n_{i-1,j,k}-\frac{3}{4}u^n_{i-1,j+1,k}\right.$$

$$\left.+\frac{1}{4}\left(u^n_{i,j+1,k}+u^n_{i-1,j+2,k}+u^n_{i,j+2,k}\right)\right]+u^n_{i,j,k}$$

$$\Leftrightarrow u^{n+1}_{i,j,k}\left(\underbrace{1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{2h_y^2}}_{c}\right)+u^{n+1}_{i,j-2,k}\left(\underbrace{\frac{-D_2\Delta t}{4h_y^2}}_{a}\right)+u^{n+1}_{i-1,j,k}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)+$$

$$+u^{n+1}_{i,j+2,k}\left(\underbrace{\frac{-D_2\Delta t}{4h_y^2}}_{a}\right)+u^{n+1}_{i+1,j,k}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)$$

$$=\underbrace{\frac{1}{2}D_1\hat{\delta}_x^2u^n_{i,j,k}+\frac{1}{2}\frac{D_2}{h_y^2}\left[\frac{1}{2}(u^n_{i,j,k}+u^n_{i,j-2,k})-2u^n_{i,j,k}+\frac{1}{2}(u^n_{i,j+2,k}+u^n_{i,j,k})\right]+u^n_{i,j,k}}_{d^n_{i,j,k}}$$

$$\Leftrightarrow au_{i,j-2,k}+bu_{i-1,j,k}+cu_{i,j,k}+bu_{i+1,j,k}+au_{i,j+2,k}=d^n_{i,j,k}$$

This procedure leads to a linear system Au = f , where

$u = [u_{2,2}\ u_{3,2}\ \ldots\ u_{n/2-1,2}\ ,\ u_{2,3}u_{3,3}\ \ldots\ u_{n/2-1,3}\ ,\ \ldots\ u_{2,n/2-1}u_{3,n/2-1}\ \ldots\ u_{n/2-1,n/2-1}\ ]T$

$f = [d_{2,2}\ d_{3,2}\ \ldots\ d_{n/2-1,2}\ ,\ d_{2,3}\ d_{3,3}\ \ldots\ d_{n/2-1,3}\ ,\ \ldots\ d_{2,n/2-1}\ d_{3,n/2-1}\ \ldots\ d_{n/2-1,n/2-1}\ ]T$

The index k + 1 is omitted in the vectors u, f for the sake of better presentation. The matrix A is given by

$$A=\left[\begin{array}{ccccccccccccccccc}
c&b&0&0&0&0&0&a&&&&&&&&&\\
b&c&b&&&&&&a&&&&&&&&\\
0&b&c&b&&&&&&a&&&&&&&\\
0&&b&c&b&&&&&&a&&&&&&\\
0&&&b&c&b&&&&&&&&&&&\\
0&&&&b&c&b&&&&&&&&&&\\
0&&&&&b&c&b&&&&&&&&&\\
a&&&&&&b&c&b&&&&&&&&\\
0&a&&&&&&b&c&b&&&&&&&\\
\vdots&&\ddots&&&&&&\ddots&\ddots&\ddots&&&&\ddots&&\\
\vdots&&&\ddots&&&&&&\ddots&\ddots&\ddots&&&&\ddots&\\
&&&&&a&&&&&b&c&b&&&&a\\
\vdots&&&&&&\ddots&&&&&\ddots&\ddots&\ddots&&&\vdots\\
&&&&&&&&a&&&&&&b&c&b\\
0&&&&&&&&&a&0&0&0&0&0&b&c
\end{array}\right]$$

For the $(i$ + 1)st step of the iterative operator splitting method, as restriction operator we choose the most obvious restriction operator, the injection. We prefer it instead of the standard full weighting operator because it leads to a linear system with better structure that is easier to handle. Injection is defined by

$$v^{2h}_{i,j}=v^{2h}_{2i,2j}.$$

Step i + 1:

$$\frac{u^{n+1}_{i+1,j,k+1}-u^n_{i+1,j,k+1}}{\Delta t}$$

$$=\frac{1}{2}\left[\left(RD_1\frac{u^{n+1}_{i-1,j,k}-2u^{n+1}_{i,j,k}+u^{n+1}_{i+1,j,k}}{h_x^2}\right.\right.$$

$$\left.+D_2\frac{u^{n+1}_{i+1,j-1,k+1}-2u^{n+1}_{i+1,j,k+1}+u^{n+1}_{i+1,j+1,k+1}}{h_y^2}\right)$$

$$+\left(RD_1\frac{u^n_{i-1,j,k}-2u^n_{i,j,k}+u^n_{i+1,j,k}}{h_x^2}\right.$$

$$\left.\left.+D_2\frac{u^n_{i+1,j-1,k+1}-2u^n_{i+1,j,k+1}+u^n_{i+1,j+1,k+1}}{h_y^2}\right)\right]$$

$$=\frac{D_1}{2}R\delta_x^2u^{n+1}_{i,j,k}+\frac{D_2}{2}\delta_y^2u^{n+1}_{i+1,j,k+1}+\frac{D_1}{2}R\delta_x^2u^n_{i,j,k}$$

$$+\frac{D_2}{2}\delta_Y^2u^n_{i+1,j,k+1}$$

$$\Leftrightarrow u^{n+1}_{i+1,j-1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+Ru^{n+1}_{i-1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i+1,j,k+1}$$

$$+Ru^{n+1}_{i,j,k}\frac{D_1\Delta t}{h_x^2}+u^{n+1}_{i+1,j,k+1}\frac{D_2\Delta t}{h_y^2}$$

$$+u^{n+1}_{i+1,j+1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+Ru^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}\left(D_1R\delta_x^2u^n_{i,j,k}+D_2\delta_y^2u^n_{i+1,j,k+1}\right)$$

$$+u^n_{i+1,j,k+1}$$

$$\Leftrightarrow u^{n+1}_{i+1,j-1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i-1,j,k+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i+1,j,k+1}$$

$$+u^{n+1}_{i,j,k+1}\frac{D_1\Delta t}{h_x^2}+u^{n+1}_{i+1,j,k+1}\frac{D_2\Delta t}{h_y^2}$$

$$+u^{n+1}_{i+1,j+1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)$$

$$+u^{n+1}_{i+1,j,k+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}\left(D_1\delta_x^2u^n_{i,j,k+1}+D_2\delta_y^2u^n_{i+1,j,k+1}\right)$$

$$+u^n_{i+1,j,k+1}$$

$$\Leftrightarrow u^{n+1}_{i,j-1,k+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u^{n+1}_{i-1,j,k+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)+u^{n+1}_{i,j,k+1}\underbrace{(1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{h_y^2})}_{c}$$

$$+u^{n+1}_{i,j+1,k+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u^{n+1}_{i+1,j,k+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)=\underbrace{\frac{1}{2}(D_1\hat{\delta}_x^2u^n_{i,j,k+1}+D_2\hat{\delta}_y^2u^n_{i+1,j,k+1})+u^n_{i,j,k+1}}_{d^n_{i,j,k+1}}$$

$$\Leftrightarrow au_{i,j-1,k+1}+bu_{i-1,j,k+1}+cu_{i,j,k+1}+bu_{i+1,j,k+1}$$

$$+au_{i,j+1,k+1}=d^n_{i,j,k+1}$$

Au = f , where

$u = [u_{2,2}\ u_{3,2}\ \ldots\ u_{n/2-1,2}\ ,\ u_{2,3}\ u_{3,3}\ \ldots\ u_{n/2-1,3}\ ,\ \ldots\ u_{2,n/2-1}\ u_{3,n/2-1}\ \ldots\ u_{n/2-1,n/2-1}\ ]T$

$f = [d_{2,2}\ d_{3,2}\ \ldots\ d_{n/2-1,2}\ ,\ d_{2,3}\ d_{3,3}\ \ldots\ d_{n/2-1,3}\ ,\ \ldots\ d_{2,n/2-1}\ d_{3,n/2-1}\ \ldots\ d_{n/2-1,n/2-1}\ ]T$

(The index k+1 is omitted in the vectors u, f for the sake of better presentation.)

$$A=\left[\begin{array}{cccccccccccccc}
c & b & 0 & 0 & a & 0 & & & & & & & & \\
b & c & b & & & a & & & & & & & & \\
0 & b & c & b & & & a & & & & & & & \\
0 & & b & c & b & & & a & & & & & & \\
a & & & b & c & b & & & a & & & & & \\
0 & a & & & b & c & b & & & a & & & & \\
 & & a & & & b & c & b & & & a & & & \\
\vdots & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & & \\
\vdots & & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & \\
 & & & & & a & & & b & c & b & & & a \\
\vdots & & & & & & \ddots & & & \ddots & \ddots & \ddots & & \vdots \\
 & & & & & & & & a & & & b & c & b \\
0 & & & & & & & & 0 & a & 0 & 0 & b & c
\end{array}\right]$$

The Jacobi iterations for the heat equation are given by

$$u_{ij}^{(n+1)}=\Delta t D_1\frac{-u_{i-1,j}^{(n+1)}+2u_{ij}^{(n+1)}-u_{i+1,j}^{(n+1)}}{h_x^2}+\Delta t D_2\frac{-u_{i,j-1}^{(n+1)}+2u_{ij}^{(n+1)}-u_{i,j+1}^{(n+1)}}{h_y^2}+u_{ij}^{(n)}$$

B. *Implementation of the Iterative Splitting Method with Embedded Two-grid Method*

The implementation for the heat equation is given by

$$\begin{aligned}
\frac{u_{i,j,k}^{n+1}-u_{i,j,k}^{n}}{\Delta t}&=\frac{1}{2}\Bigg[\Bigg(D_1\frac{u_{i-1,j,k}^{n+1}-2u_{i,j,k}^{n+1}+u_{i+1,j,k}^{n+1}}{h_x^2}\\
&\quad+PD_2\frac{u_{i,j-1,k-1}^{n+1}-2u_{i,j,k-1}^{n+1}+u_{i,j+1,k-1}^{n+1}}{h_y^2}\Bigg)\\
&\quad+\Bigg(D_1R\frac{u_{i-1,j,k}^{n}-2u_{i,j,k}^{n}+u_{i+1,j,k}^{n}}{h_x^2}\\
&\quad+D_2\frac{u_{i,j-1,k+1}^{n}-2u_{i,j,k+1}^{n}+u_{i,j+1,k+1}^{n}}{h_y^2}\Bigg)\Bigg]\\
&=\frac{D_1}{2}\delta_x^2u_{i,j,k}^{n+1}+\frac{D_2}{2}P\delta_y^2u_{i,j,k-1}^{n+1}+\frac{D_1}{2}R\delta_x^2u_{i,j,k}^{n}\\
&\quad+\frac{D_2}{2}\delta_Y^2u_{i,j,k+1}^{n}
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow Pu_{i,j-1,k-1}^{n+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)&+Ru_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u_{i,j,k}^{n+1}\\
&+u_{i,j,k}^{n+1}\frac{D_1\Delta t}{h_x^2}+Pu_{i,j,k-1}^{n+1}\frac{D_2\Delta t}{h_y^2}\\
&+Pu_{i,j+1,k-1}^{n+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+Ru_{j+1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)\\
&+u_{i,j,k}^{n+1}\\
&=\frac{1}{2}\left(D_1R\delta_x^2u_{i,j,k}^{n}+D_2P\delta_y^2u_{i,j,k+1}^{n}\right)+u_{i,j,k}^{n}
\end{aligned}$$

$$\Leftrightarrow u_{i,j-1,k}^{n+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u_{i-1,j,k-1}^{n+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)+u_{i,j,k}^{n+1}\underbrace{(1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{h_y^2})}_{c}$$

$$+u_{i,j+1,k}^{n+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u_{i+1,j,k+1}^{n+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)=\underbrace{\frac{1}{2}(D_1\hat{\delta}_x^2u_{i,j,k+1}^{n}+D_2\hat{\delta}_y^2u_{i,j,k+1}^{n})+u_{i,j}^{n}}_{d_{i,j}^{n}}$$

$$\Leftrightarrow au_{i,j-1,k}^{n+1}+bu_{i-1,j,k-1}^{n+1}+cu_{i,j,k}^{n+1}+au_{i,j+1,k}^{n+1}+bu_{i+1,j,k+1}^{n+1}=d_{i,j}^{n}\qquad(1)$$

The restriction operators are given by

Ru_u^{n+1} lead to $\frac{1}{16}\left[u_{i-1,j-1,k-1}^{n+1}+2u_{i,j-1,k-1}^{n+1}+u_{i+1,j-1,k-1}^{n+1}+2u_{i-1,j,k-1}^{n+1}+4u_{i,j,k-1}^{n+1}+2u_{i+1,j,k-1}^{n+1}+u_{i-1,j+1,k-1}^{n+1}+2u_{i,j+1,k-1}^{n+1}+u_{i+1,j+1,k-1}^{n+1}\right]$,

according to the interpretation of the stencil

$$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

V. NUMERICAL EXPERIMENTS

In the following examples, we deal with different test examples and their underlying multi-scale physics.

A. *Simple Heat Equation*

We deal with a PDE which is parabolic and has stiffness in the diffusion part.

We have the following equation:

$$\partial_t u_1=D_{11}\,\partial_{xx}u_1+D_{21}\,\partial_{xx}u_2,$$

$$\partial_t u_2=D_{12}\,\partial_{xx}u_1+D_{22}\,\partial_{xx}u_2,\qquad(80)$$

$$u_1(0)=u_{10},u_2(0)=u_{20}(\text{initial conditions}),\qquad(81)$$

$$u_1(x,t)=\;u_2(x,t)=0\;(\text{boundary conditions}),\qquad(82)$$

where D11 , D21 , D12 , D22 $\in$ IR+ and D11 , D12 < D21, D22 are the diffusion operators.

We apply the finite difference scheme for the spatial derivatives:

$\partial xx=[1-2\ 1]$ and $\partial yy=[1-2\ 1]t$.

With the standard pro jection and restriction:

$$R_H=1/16\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

$$P_h=4R_H$$

The time-derivations are discretised by implicit Euler methods and we obtain the following system of linear equations:

$$(c_i^{n+1}-c_i^{n})/\Delta t=Ac_i^{n+1}+PBc_{i-1}^{n+1},\text{with }c_i(t^n)=c^n\qquad(83)$$

$$(c_{i+1}^{n+1}-c_{i+1}^{n})/\Delta t=RAc_i^{n+1}+Bc_{i+1}^{n+1},$$

$$\text{with }c_i+1(t^n)=\;c^n\qquad(84)$$

We obtain, after the insertion of the operators, a system of linear equations:

$$AU_i^{n+1}=BU_{i-1}^{n+1}+\;CU_i^{n},\qquad(85)$$

This is solved by a two-grid method. Here we have two scales and decouple:

$$\partial_t u=Au+Bu,\qquad(86)$$

$$u(0)=(u_{10},u_{20})^T,\qquad(87)$$

where $u(t)=(u1\ (t),u2\ (t))^T$ for $t\in[0,T]$.
Our split operators are

$$A = \begin{pmatrix} D_{11}\,\partial_{xx} & 0 \\ D_{12}\,\partial_{xx} & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & D_{21}\,\partial_{xx} \\ 0 & D_{22}\,\partial_{xx} \end{pmatrix}. \quad (88)$$

We chose such an example to have $AB \neq BA$, and therefore we have a splitting error of first order for the usual sequential splitting methods, called A–B splitting.

Our numerical results based on two-grid methods are presented in the following Table Ⅰ.

We choose D11 = D12 = 0.5 and D21 = D22 = 0.05 on the time interval [0, 1]. As second order method we choose Crank-Nicolson with $\theta = 1/2$. As fourth order method we choose the Gauss Runge-Kutta. We apply a two-grid method, see subsection Ⅳ-A.

The numerical results are given in Table Ⅰ. We apply the L_2 error based on the exact and the numerical solution.

TABLE Ⅰ NUMERICAL RESULTS FOR THE FIRST EXAMPLE WITH THE ITERATIVE SPLITTING METHOD AND SECOND AND FOURTH ORDER METHODS

Number of time-partitions	Iterative Steps	err_1 (order 2)	err_2 (order 2)	err_1 (order 4)	err_2 (order 4)
2	1	4.5321e-002	3.6077e-003	4.5321e-002	3.6077e-003
2	10	3.9664e-003	4.7396e-004	3.9664e-003	4.7397e-004
2	100	3.9204e-004	4.8078e-005	3.9204e-004	4.8083e-005
3	1	4.6126e-004	3.6077e-003	4.6126e-004	3.6077e-003
3	10	7.8129e-006	2.9285e-005	7.8069e-006	2.9289e-005
3	100	8.5988e-008	2.8270e-007	8.0050e-008	2.8682e-007
4	1	4.6126e-004	2.2459e-005	4.6126e-004	2.2464e-005
4	10	4.1883e-007	4.2629e-008	4.1321e-007	4.8154e-008
4	100	5.9521e-009	5.4846e-009	4.0839e-010	4.9968e-011

Remark 4. In the experiments, we improved the iterative splitting scheme with higher order discretisation methods. Further we can accelerate the splitting scheme with an underlying two-grid method that is embedded in the splitting scheme. Both higher order time-discretisation and also two-grid methods are necessary to obtain such results.

B. Second Example: Transport–Reaction Equation

We deal with the following system of transport equations:

$$\partial_t u_1 + v_1\,\partial_x u_1 = -\lambda u_1 + \mu u_2, \quad (89)$$

$$\partial_t u_2 + v_2\,\partial_x u_2 = \mu u_1 - \lambda u_2, \quad (90)$$

where we have Dirichlet boundary conditions and $u_{1,0}$, $u_{2,0}$ are the initial conditions.

We split the operator into two operators:

$$A = \begin{pmatrix} -v_1\,\partial_x & \\ & -v_2\,\partial_x \end{pmatrix}, \quad B = \begin{pmatrix} -\lambda & \mu \\ \mu & -\lambda \end{pmatrix} \quad (91)$$

We see that for $\mu \approx 0$ the operators commute.
We choose $v1 = 1$, $v2 = 0.5$, $\lambda = 0.01$. We let $t \in [0, 40]$ and $x \in [0, 40]$.
We set

$$\Delta t = \frac{1}{25} \text{ and } \Delta x = \frac{4}{10}$$

In Fig. 2, we choose $\mu = 0.001$ and see that we could say $\mu \approx 0$ and obtain nearly the same results as for the A–B splitting.

In Fig. 3, we choose $\mu = 0.01$ and see that $\mu = 0$ and obtain more optimal results for the iterative schemes.

The result is given in Figs. 2-4.

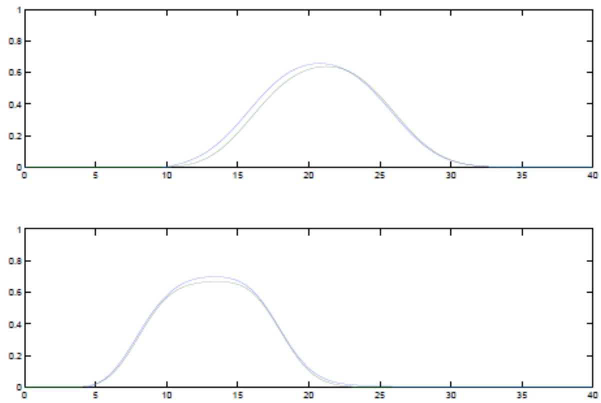

Fig. 2 Blue - Iterative Operator Splitting (4 iterations); Green - A–B Splitting

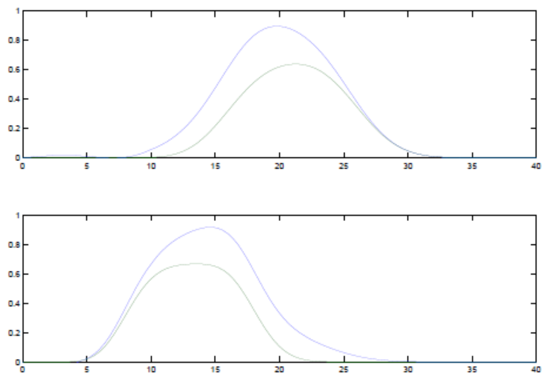

Fig. 3 T=2: Blue - Iterative Operator Splitting (4 iterations); Green - A–B Splitting

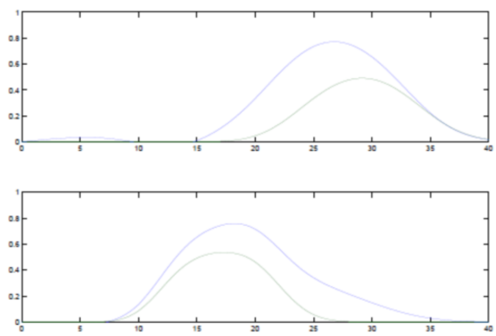

Fig. 4 T=60: Blue - Iterative Operator Splitting (4 iterations); Green - A–B Splitting

Remark 5. In this example, we concentrate on the comparison between the standard A–B splitting and the iterative splitting methods. We obtain more accurate results for non-commuting problems. Such problems are related to multi-scale problems and we can apply our embedded multi-grid method. We have the benefit of receiving higher order results without reducing the time-step.

VI. CONCLUSIONS AND DISCUSSION

We presented iterative splitting methods with embedded multi-grid and multi-stepping methods. Here the idea was to achieve more accurate and faster results than the standard splitting schemes (e.g., the A–B splitting schemes). We

obtained a benefit from this embedded method in accelerating the iterative schemes and achieved much more accurate results. In the future we will concentrate on nonlinear and matrix dependent splitting schemes.

REFERENCES

[1] W. Bangerth and R. Rannacher. Adaptive Finite Element Methods for Differential Equations. Lectures in Mathematics, ETH Zu ̈rich, Birkh ̈auser Verlag, Basel, 2003.

[2] D. Braess. Finite Elemente. Springer-Verlag, Berlin, 1992.

[3] X.C. Cai, Additive Schwarz algorithms for parabolic convection–diffusion equations, Numer. Math. 60, 41–61, 1991.

[4] X.C. Cai, Multiplicative Schwarz methods for parabolic problems, SIAM J. Sci Comput. 15, 587–603, 1994.

[5] W. Cheney, Analysis for Applied Mathematics, Graduate Texts in Mathematics, vol. 208, Springer-Verlag, New York, 2001.

[6] C.N. Dawson, Q. Du, and D.F. Dupont, A finite difference domain decomposition algorithm for numerical solution of the heat equation, Mathematics of Computation 57, 63–71, 1991.

[7] P. Deuflhard, Newton Methods for Nonlinear Problems, Springer-Verlag, Berlin, 2004.

[8] K.-J. Engel and R. Nagel, One-Parameter Semigroups for Linear Evolution Equations, Springer, New York, 2000.

[9] I. Farago and J. Geiser, Iterative Operator-Splitting methods for Linear Problems, Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, Berlin. International Journal of Computational Science and Engineering, accepted September 2007.

[10] M.J. Gander and H. Zhao, Overlapping Schwarz waveform relaxation for parabolic problems in higher dimension. In: A. Handlovi ̌cova ́, Magda Komorn ́ıkova, and Karol Mikula (eds.), Proc. Algoritmy 14, Slovak Technical University, 1997, pp.42–51.

[11] E. Giladi and H. Keller, Space time domain decomposition for parabolic problems. Technical Report 97-4, Center for Research on Parallel Computation CRPC, Cal- tech, 1997.

[12] J. Geiser, Discretisation with embedded analytical solutions for convection domi- nated transport in porous media. In: Proc. NA&A ’04, Lecture Notes in Computer Science, Vol. 3401, Springer, Berlin, 2005, pp. 288–295.

[13] J. Geiser, Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations, J. Comput. Appl. Math., accepted, June 2007.

[14] J. Geiser, O. Klein, and P. Philip. Influence of anisotropic thermal conductivity in the apparatus insulation for sublimation growth of SiC: Numerical investigation of heat transfer. Crystal Growth & Design 6, 2021–2028, 2006.

[15] J. Geiser. Decomposition Methods for Differential Equations: Theory and Applications. CRC Press, Numerical Analysis and Scientific Computing Series, edited by Magoules and Lai, 2009.

[16] J. Geiser and M. Arab. Modelling and simulation of a chemical vapor deposition. Journal of Applied Mathematics, special issue: Mathematical and Numerical Modeling of Flow and Transport (MNMFT), Hindawi Publishing Corp., New York, accepted, January 2011.

[17] W. Hackbusch. Multi-Grid Methods and Applications. Springer-Verlag, Berlin, 1985.

[18] W. Hackbusch. Iterative L ̈osung gro ̈ser schwachbesetzter Gleichungssysteme. Teubner Studienbu ̈cher: Mathematik, B.G. Teubner Stuttgart, 1993.

[19] E. Hannsen and A. Ostermann, Dimensional splitting for evolution equations.Numer. Math.,108, 557–570, 2008.

[20] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2003.

[21] M. Holst, R. Kozack, F. Saied and S. Subramaniam, Treatment of Electrostatic Effects in Proteins: Multigrid-based Newton Iterative Method for Solution of the Full Nonlinear Poisson–Boltzmann Equation, Proteins: Stucture, Function, and Genetics, vol. 18, 231–245, 1994.

[22] W. Hundsdorfer and J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Series in Computational Mathematics, Vol. 33, Springer-Verlag, Berlin, 2003.

[23] R. Jeltsch and O. Nevanlinna, Stability and accuracy of time discretizations for initial value problems. Numerische Mathematik, 40(2), 245–296, 1982.

[24] J. Kanney, C. Miller, and C.T. Kelley, Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems, Advances in Water Resources 26, 247–261, 2003.

[25] K.H. Karlsen and N. Risebro. An operator splitting method for nonlinear convection–diffusion equation. Numer. Math., 77(3), 365–382, 1997.

[26] K.H. Karlsen and N.H. Risebro, Corrected operator splitting for nonlinear parabolic equations, SIAM J. Numer. Anal. 37, 980–1003, 2000.

[27] K.H. Karlsen, K.A. Lie, J.R. Natvig, H.F. Nordhaug and H.K. Dahle, Operator splitting methods for systems of convection–diffusion equations: Nonlinear error mechanisms and correction strategies, J. Comput. Phys. 173, 636–663, 2001.

[28] C.T. Kelly. Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 1995.

[29] P. Knabner and L. Angermann, Numerical Methods for Elliptic and Parabolic Partial Differential Equations, Texts in Applied Mathematics, vol. 44, Springer-Verlag, Berlin, 2003.

[30] M.A.Lieberman and A.J. Lichtenberg. Principle of Plasma Discharges and Materials Processing, Wiley-Interscience, Second Edition, 2005.

[31] G.I. Marchuk, Some applicatons of splitting-up methods to the solution of problems in mathematical physics, Aplikace Matematiky 1, 103–132, 1968.

[32] G.A. Meurant, Numerical experiments with a domain decomposition method for parabolic problems on parallel computers. In: R. Glowinski, Y.A. Kuznetsov, G.A. Meurant, J. P ́eriaux and O. Widlund, (eds.), Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA, 1991. SIAM.

[33] U Miekkala and O. Nevanlinna. Sets of convergence and stability regions. BIT Numerical Mathematics, 27, 554–584, 1987.

[34] U Miekkala and O. Nevanlinna. Convergence of dynamic iteration methods forinitial value systems. SIAM, 8(4), 459–482, 1987.

[35] H.A. Schwarz, ̈Uber einige Abbildungsaufgaben, Journal f ̈ur Reine und Ange-wandte Mathematik 70, 105–120, 1869.

[36] H. Roos, M. Stynes and L. Tobiska. Numerical Methods for Singular Perturbed Differential Equations, Springer-Verlag, Berlin, 1996.

[37] H.A. Schwarz, ̈Uber einige Abbildungsaufgaben, Journal f ̈ur Reine und Ange-wandte Mathematik 70, 105–120, 1869.

[38] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5, 506–517, 1968.

[39] A. Quarteroni and A. Valli. Numerical Approximation of Partial Deferential Equations Springer Series in Computational Mathematics, Springer-Verlag, Berlin,1997.

[40] A. Quarteroni and A. Valli. Domain Decomposition Methods for Partial Differential Equations Series: Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1999.

[41] S. Vandewalle. Parallel Multigrid Waveform Relaxation for Parabolic Problems. Teubner, Stuttgart, 1993.

[42] 42. H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2003.

[43] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A, Vol. 150, nos. 5,6,7, 1990.

[44] H. Yserentant. Old and New Convergence Proofs for Multigrid Methods. Acta Numerica, 285–326, 1993.

[45] E. Zeidler. Nonlinear Functional Analysis and its Applications. II/B Nonlinear montone operators Springer-Verlag, Berlin, 1990.

[46] Z. Zlatev. Computer Treatment of Large Air Pollution Models. Kluwer Academic Publishers, 1995.